Amendments to the Claims

Please amend Claims 1, 5, 9, and 13. The Claim Listing below will replace all prior versions of the claims in the application:

Claim Listing

1. (Currently amended) A method for updating a lookup table comprising the steps of:

providing access to a first set of routes stored in nodes of a first subtree within a tree, the first subtree being accessed through a first pointer to a first subtree root node of the first subtree;

providing a second subtree disconnected from the tree, the second subtree being accessed through a second pointer to a second subtree root node of the second subtree and being initially inaccessible via the tree;

storing a second set of routes in nodes of the second subtree while access via the tree is provided to the first set of routes stored in the first subtree by the first pointer, the second set of routes being ~~derived from~~ a copy of the first set of routes with either a new route added or an existing route deleted; and

switching access to the second set of routes stored in the second subtree by replacing the first pointer to the first subtree root node with the second pointer to the second subtree root node to update the tree by replacing the first subtree with the second subtree.

2. (Previously presented) The method as claimed in Claim 1 further comprising the step of:

deallocating memory used by the first set of routes after switching access.

3. (Original) The method as claimed in Claim 1 wherein the number of routes in the first set of routes is less than the number of routes in the second set of routes.

4. (Original) The method as claimed in Claim 1 wherein the number of routes in the first set of routes is greater than the number of routes in the second set of routes.

5. (Currently amended) An apparatus for updating a lookup table comprising:

a first subtree within a tree, the first subtree including a first subtree root node and a first set of routes stored in nodes of the first subtree, the first set of routes being initially accessible via the tree;

a first pointer to the first subtree root node;

a second subtree initially disconnected from the tree, the second subtree including a second subtree root node and a second set of routes stored in nodes of the second subtree, the second set of routes being ~~derived from~~ a copy of the first set of routes with either a new route added or an existing route deleted and being initially inaccessible via the tree;

a second pointer to the second subtree root node; and

logic to switch access to the second set of routes by replacing in the tree the first pointer to the first subtree root node with the second pointer to the second subtree root node to update the tree by replacing the first subtree with the second subtree.

6. (Previously presented) The apparatus as claimed in Claim 5 further comprising:

deallocation logic to deallocate memory used by the first set of routes after switching access.

7. (Original) The apparatus as claimed in Claim 5 wherein the number of routes in the first set of routes is less than the number of routes in the second set of routes.

8. (Original) The apparatus as claimed in Claim 5 wherein the number of routes in the first set of routes is greater than the number of routes in the second set of routes.

9. (Currently amended) An apparatus for updating a lookup table comprising:

a first pointer to a first subtree root node of a first subtree within a tree, the first pointer providing access to a first set of routes stored in nodes of the first subtree;

a second pointer to a second subtree root node of a second subtree disconnected from the tree, the second pointer providing access to a second set of routes stored in nodes of the second subtree, the second set of routes being ~~derived from~~ a copy of the first set of routes with either a new route added or an existing route deleted and being initially inaccessible via the tree while access via the tree is provided to the first set of routes stored in the first subtree by the first pointer; and

logic to provide access to the second set of routes by replacing the first pointer to the first subtree root node with the second pointer to the second subtree root node to update the tree by replacing the first subtree with the second subtree.

10. (Previously presented) The apparatus as claimed in Claim 9 further comprising:

deallocation logic to deallocate memory used by the first set of routes after the first pointer is replaced.

11. (Original) The apparatus as claimed in Claim 9 wherein the number of routes in the first set of routes is less than the number of routes in the second set of routes.

12. (Original) The apparatus as claimed in Claim 9 wherein the number of routes in the first set of routes is greater than the number of routes in the second set of routes.

13. (Currently amended) A method for updating a lookup table, the lookup table providing a longest prefix match for a destination address, comprising the steps of:

providing access to a first set of routes stored in nodes of a first subtree within a tree, the first subtree being accessed through a first pointer to a first subtree root node of the first subtree;

providing a second subtree disconnected from the tree, the second subtree being accessed through a second pointer to a second subtree root node of the second subtree and being initially inaccessible via the tree;

storing a second set of routes in nodes of the second subtree while access via the tree is provided to the first set of routes stored in the first subtree by the first pointer, the

second set of routes being ~~derived from~~ a copy of the first set of routes with either a new route added or an existing route deleted; and

switching access to the second set of routes stored in the second subtree by replacing the first pointer to the first subtree root node with the second pointer to the second subtree root node to update the tree by replacing the first subtree with the second subtree.

14. (Previously presented) The method of claim 13, wherein the first set of routes and the second set of routes include a longest prefix route for the destination address.

15. (Previously presented) The method of claim 14, wherein the destination address includes an Internet Protocol address.

16. (Previously presented) The method of claim 14, wherein the second set of routes includes another route corresponding to the longest prefix route for another destination address.

17. (Previously presented) The method of claim 13, wherein the first set of routes and the second set of routes are associated with nodes at the bottom level of a subtree.

18. (Previously presented) A method as claimed in claim 1 wherein storing the second set of routes in the nodes of the second subtree includes storing a copy of the first set of routes in the nodes of the second subtree and storing at least one additional route in the nodes of the second subtree.

19. (Previously presented) An apparatus as claimed in claim 5 wherein the second set of routes includes a copy of the first set of routes and includes at least one additional route.

20. (Previously presented) An apparatus as claimed in claim 9 wherein the second set of routes includes a copy of the first set of routes and includes at least one additional route.